UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/971,720	10/04/2001	David Ian Houlding	92717-315	3044

7590 12/15/2006

Gary B. Solomon
Jenkens & Gilchrist, P.C.
3200 Fountain Place
1445 Ross Avenue
Dallas, TX 75202-2799

EXAMINER	
FOWLKES, ANDRE R	

ART UNIT	PAPER NUMBER
2192	

DATE MAILED: 12/15/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/971,720	HOULDING, DAVID IAN	
	Examiner	Art Unit	
	Andre R. Fowlkes	2192	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *04 October 2006*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-5,7,8,11,13-15,21,22,24-26,32,34,41 and 45-47* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-5, 7-8, 11, 13-15, 21-22, 24-26, 32, 34, 41 & 45-47* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. This action is in response to the amendment filed 10/4/06.

2. Claims 1-5, 7-8, 11, 13-15, 21-22, 24-26, 32, 34, 41 & 45-47 are pending. No claims have been amended, canceled or added.

Claim Rejections - 35 USC § 103

3. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

4. Claims 1-5, 7-8, 11, 13-15, 21-22, 24-26, 32, 34, 41 & 45-47 are rejected under 35 U.S.C. 103(a) as being unpatentable over Beatty et al. (Beatty), U.S. Patent No. 5,913,052 in view of Weinberg et al. (Weinberg) U.S. Patent No. 6,144,962.

As per claim 1, Beatty discloses a system and method for debugging software with an architectural view **within a network** (col. 5:12-19, "It should be noted that any conventional computer system having at least one processor that is suitable to function as a general purpose computer may replace, or be used in conjunction with, the computer 100, including, without limitation: hand-held, laptop/notebook, mini, mainframe

and supercomputers, including RISC and parallel processing architectures, as well as within computer system/network combinations."), and in that, Beatty covers the steps of:

- accessing a datafile descriptive of the underlying architecture (col. 6 lines 20-23, "the memory stores (a datafile containing) a plurality of user-selectable architectures corresponding to a plurality of (underlying architectures)"),

- transforming the datafile to determine architectural components used to form the underlying architecture and rendering, via a visualizer, a plurality of graphical elements representative of the architectural components (col. 6 lines 23-26, "(The datafile is transformed and then) the display coupled to the memory (i.e. the visualizer), displays (the architectural components of the underlying architecture)"),

- the graphical elements forming a graphical representation of the underlying architecture (col. 6 lines 24-25, "a window on the display (includes a graphical representation of the underlying) architecture including a graphical device layout", and col. 9:16-19, "As represented in another higher resolution illustration of the architectural window 410, all of the corresponding states of the registers in the DSP pipeline (i.e. graphical representation of the underlying architecture of the software system), have been updated to represent the latest instructions." Beatty further discloses updating the graphical representation in the transition from instruction 28 to 29 as shown in figures 8 and 9. In the architectural display window (top left) of figure 8, the arithmetic logic units (the upside-down trapezoid shaped objects, marked X) have output registers with undefined/empty values. After transitioning to step 29, in figure 9, the output registers of both arithmetic logic units have been updated to show that they

are full and have the specific values of pah = 0x1bb0, pal = 0xd9c4, pbh = 0x1bb1 and pbl = 0x8229),

- the graphical representation dependant on a particular mode of a plurality of modes of operation of the visualizer (Beatty describes how the graphical representation is dependent on a particular mode of a plurality of modes of operation at, col. 3:28-3:43, "In an alternative embodiment of the present invention, the system further comprises source software display circuitry that displays a source code representation of the DSP software in a further window on the display ... In an alternative embodiment of the present invention, the system further comprises object software display circuitry that displays an object code representation of the DSP software in a further window on the display"),

- providing at least one control on the webpage **and receiving a selection of the at least one control** (col. 6 lines 26-30, "The processor, coupled to the display (i.e. the webpage) (provides controls to) simulate operation of the ... software and emulated operation of the (underlying architecture) ... to cause the (underlying architecture) to change states"),

- performing a graphical operation on the webpage **for dynamic visualization of the graphical elements indicative of the underlying architecture** (col. 6 lines 30-31, "The processor controls the display to update the (graphical display of the underlying architecture in ways indicative of its dynamic functional operations)").

- wherein the dynamic visualization provides a graphical representation of collaborative interactions between the architectural components of the

underlying architecture of the software system (col. 2 lines 39-49, "When the DSP is emulated, the system includes: (1) architectural display circuitry that displays an architecture of a particular DSP in a window on a display of the general purpose computer, the architecture including a graphical device layout (i.e. an architectural component that interacts with the register of the DSP) and at least one field corresponding to a register of the DSP (i.e. an architectural component) and (2) software simulation circuitry that employs a processor of the general purpose computer to simulate operation of DSP software (i.e. an architectural component) and emulate operation of the particular DSP to cause the particular DSP (i.e. an architectural component) to change states over time (through collaborative interaction), the architectural display circuitry updating the at least one field to reflect changes in the states", and col. 9:16-19, "As represented in another higher resolution illustration of the architectural window 410, all of the corresponding states of the registers in the DSP pipeline (i.e. graphical representation of the underlying architecture of the software system), have been updated to represent the latest instructions." Beatty further discloses updating the graphical representation in the transition from instruction 28 to 29 as shown in figures 8 and 9. In the architectural display window (top left) of figure 8, the arithmetic logic units (the upside-down trapezoid shaped objects, marked X) have output registers with undefined/empty values. After transitioning to step 29, in figure 9, the output registers of both arithmetic logic units have been updated to show that they are full and have the specific values of pah = 0x1bb0, pal = 0xd9c4, pbh = 0x1bb1 and pbl = 0x8229. This displays the collaborative interaction between the ALU and the

register (i.e. architectural components of the underlying architecture of the software system)),

- communicating the rendered graphical representation across the network (col. 5 lines 12-20, "It should be noted that (this invention may communicate the representation across)... computer system/network combinations"),

- and wherein the visualizer is utilized for visualizing using the webpage **the underlying architecture of the software system during conceptual development and deployment phases of the software system** (col. 3:21-23, "Additionally, the present invention allows one or more of the user-selectable architectures correspond to DSPs that do not even exist (i.e. during conceptual development)", and col. 3:6-16, "the system of the present invention may include an interface between a processor of the general purpose computer and a real (non-emulated) DSP. In this alternative embodiment, the general purpose computer causes the DSP software to execute within the real DSP (i.e. deployment phase)"),

Beatty doesn't explicitly disclose **displaying, on a web page,** the graphical representation of the underlying architecture of the software system.

However, Weinberg, in an analogous environment, discloses **displaying, on a web page,** the graphical representation of the underlying architecture of the software system (Fig. 11, and associated text (e.g. col. 23 line 39 – col. 24 line 25), show that the graphical display is a web page).

Therefore, it would have been obvious to a person of ordinary skill in the art, at the time the invention was made, to incorporate the teachings of Weinberg into the system of Beatty to **display, on a web page,** the graphical representation of the underlying architecture of the software system. The modification would have been obvious because one of ordinary skill in the art would have wanted a convenient, universal way to display the graphical information to the user (Beatty, col. 2 lines 27-30).

As per claim 2, the rejection of claim 1 is incorporated and further, Beatty discloses **generating a plurality of subsections of a graphical image** (Fig. 4, item 430 and the associated text, (e.g. col. 7 line 11 - col. 8 line 11), describe vertical subsections of the graphical image), **and locating the graphical elements in the subsections as described by the datafile** (col. 2 lines 42-43, "the architecture including a graphical device layout (containing the graphical elements in subsections)").

As per claim 3, the rejection of claim 1 is incorporated and further, Beatty discloses that **the subsections are displayed as tiers** (Fig. 4, item 430 and the associated text, (e.g. col. 7 line 11 - col. 8 line 11), describe the subsections of the system displayed as vertical tiers).

As per claim 4, the rejection of claim 1 is incorporated and further, Beatty discloses **providing access to the visualization on a network** (col. 5 lines 12-20, "It

should be noted that (this invention may be used with)... computer system/network combinations").

As per claim 5, the rejection of claim 4 is incorporated and further, Beatty discloses that **the network is the Internet** (col. 5 lines 12-20, "It should be noted that (this invention may be used with)... computer system/network combinations (i.e. the Internet)").

As per claim 7, the rejection of claim 1 is incorporated and further, Beatty discloses **receiving data for said rendering from a network connection** (col. 5 lines 12-20, "It should be noted that (this invention may receive data for rendering from)... computer system/network combinations").

As per claim 8, the rejection of claim 7 is incorporated and further, Beatty discloses **storing the data** (Fig. 1, item 180, and the associated text (e.g. col. 4 line 41 - col. 2 line 34), show a memory used to store data).

As per claim 11, the rejection of claim 1 is incorporated and further, Beatty doesn't explicitly disclose that **the datafile includes extensible markup language (XML).**

However, Weinberg, in an analogous environment, discloses that **the datafile includes extensible markup language (XML)** (col. 2 lines 10-13, "the program

includes Web site scanning routines ... to gather information about ... HTML (and XML) documents and links of a Web site").

Therefore, it would have been obvious to a person of ordinary skill in the art, at the time the invention was made, to incorporate the teachings of Weinberg into the system of Beatty to have the datafile include extensible markup language. The modification would have been obvious because one of ordinary skill in the art would have wanted the system to understand XML, to be able to collect as much information as possible from the data file, to analyze the software to the fullest extent, in order to enable the maximum optimization of the software and underlying architecture (Beatty, col. 2 lines 27-30).

As per claim 13, the rejection of claim 1 is incorporated and further, Beatty discloses **altering the graphical elements based on a selected configuration of the software system** (col. 6 lines 18-25, "the memory stores a plurality of ... (different graphical elements/systems representative of underlying architecture components and systems) corresponding to (a plurality of software system configurations)", and the graphical elements displayed are altered when a user chooses a different configuration).

As per claim 14, the rejection of claim 1 is incorporated and further, Beatty discloses **receiving an event initiated by an operation performed in a second graphical display operating in isolation of actual components of the underlying**

architecture (Fig. 4, items 460, 470, and 450 show additional graphical displays used to initiate events, and col. 6 lines 26-33, "The processor, couples to the display, simulates operation of the ... software and emulates operation of the particular (underlying architecture) ... to cause the particular (underlying architecture) to change states"), and **performing an operation on the graphical display based on the event** (col. 6 lines 30-33, "The processor controls the display to update the (representation of the underlying architecture)").

As per claim 15, the rejection of claim 1 is incorporated and further, Beatty discloses **receiving an event initiated by an operation performed in a second graphical display operating in conjunction with actual components of the underlying architecture** (Fig. 4, items 460, 470, and 450 show additional graphical displays used to initiate events, and col. 6 lines 34-43, "the system of the present invention my include an interface and a real (underlying architecture) ... associated with the computer. In this alternative, the processor causes the ... software to execute within the real (underlying architecture) ... to change states.)"), and **performing an operation on the graphical display based on the event** (col. 6 lines 39-43, "The processor controls the display to update the (graphical representation of the underlying architecture").

As per claims 21, 22 and 24-26, Beatty/Weinberg also discloses such claimed limitations as addressed in claims 9, 14, 15 and 11 above, respectively.

As per claim 32 and 34, these are a product version of the claimed method discussed above in claims 1 and 5, wherein all claimed limitations have also been addressed above and such a product is deemed to be inherent in the Beatty/Weinberg system and method for debugging software, otherwise it would be inoperative.

As per claim 41, Beatty discloses an **application service provider (ASP) system for visualizing an architecture of another distinct system** (col. 3:14-19, "In an alternative embodiment of the present invention, the system (i.e. ASP) further comprises an architecture database, storable on a storage device of the general purpose computer, that contains a plurality of user-selectable architectures corresponding to a plurality of DSPs (i.e. another distinct software system), the system thereby allowing the user to select the particular DSP (to visualize) from the database"),

- the ASP system comprising:

- a datafile including a description of the architecture (col. 6 lines 20-23, "the memory stores (a datafile containing) a plurality of user-selectable architectures corresponding to a plurality of (underlying architectures)"),

- a host computing system for transforming the datafile, a visualizer for receiving the transformed datafile and visualizing the architecture (col. 6 lines 23-26, "(The datafile is transformed by the host system and then) the display coupled to the memory (i.e. the visualizer), displays (i.e. visualizes the architecture)"),

- the visualizer operating in one of a plurality of modes of operation (Beatty describes how the graphical representation is dependent on a particular mode of a plurality of modes of operation at, col. 3:28-3:43, "In an alternative embodiment of the present invention, the system further comprises source software display circuitry that displays a source code representation of the DSP software in a further window on the display ... In an alternative embodiment of the present invention, the system further comprises object software display circuitry that displays an object code representation of the DSP software in a further window on the display"),

- a visual display for receiving and displaying the visualized architecture of said another distinct system (col. 6 lines 24-26, "the (visual) display coupled to the memory (i.e. the visualizer), (receives and) displays (the visualized architecture)", and col. 3:6-16, "the system of the present invention may include an interface between a processor of the general purpose computer and a real (non-emulated) DSP. In this alternative embodiment, the general purpose computer causes the DSP software to execute within the real DSP (i.e. direct interaction simulation mode), in turn causing the real DSP to change states over time. The general purpose computer monitors the states allowing the architectural display circuitry to update at least one field to reflect changes in the states", and col. 3:21-23, "Additionally, the present invention allows one or more of the user-selectable architectures correspond to DSP (software systems) that do not even exist (i.e. a distinct system)").

- wherein the visualizer is utilized for visualizing the architecture of the system during conceptual development and deployment phases of the system

(col. 3:21-23, "Additionally, the present invention allows one or more of the user-selectable architectures correspond to DSPs that do not even exist (i.e. during conceptual development)", and col. 3:6-16, "the system of the present invention may include an interface between a processor of the general purpose computer and a real (non-emulated) DSP. In this alternative embodiment, the general purpose computer causes the DSP software to execute within the real DSP (i.e. deployment phase)"),

- **wherein the visual display includes at least one control** (col. 6 lines 26-30, "The processor, coupled to the display (i.e. the webpage) (provides controls to) simulate operation of the ... software and emulated operation of the (underlying architecture) ... to cause the (underlying architecture) to change states"),

- **wherein the at least one control is adapted to perform a graphical operation on the** webpage **for dynamic visualization of architectural components indicative of an underlying architecture of the software system** (col. 6 lines 26-31, "The processor, coupled to the display (i.e. the webpage) (provides controls to) simulate operation of the ... software and emulated operation of the (underlying architecture) ... to cause the (underlying architecture) to change states. The processor controls the display to update the (graphical display of the underlying architecture in ways indicative of its dynamic functional operations)" and col. 9:16-19, "As represented in another higher resolution illustration of the architectural window 410, all of the corresponding states of the registers in the DSP pipeline (i.e. graphical representation of the underlying architecture of the software system), have been updated to represent the latest instructions." Beatty further discloses updating the graphical representation in the

transition from instruction 28 to 29 as shown in figures 8 and 9. In the architectural display window (top left) of figure 8, the arithmetic logic units (the upside-down trapezoid shaped objects, marked X) have output registers with undefined/empty values. After transitioning to step 29, in figure 9, the output registers of both arithmetic logic units have been updated to show that they are full and have the specific values of pah = 0x1bb0, pal = 0xd9c4, pbh = 0x1bb1 and pbl = 0x8229. This displays the collaborative interaction between the ALU and the register (i.e. architectural components of the underlying architecture of the software system)),

Beatty doesn't explicitly disclose that **the visual display is a webpage on the Internet**.

However, Weinberg, in an analogous environment, discloses that **the visual display is a webpage on the Internet** (Fig. 11, and associated text (e.g. col. 23 line 39 – col. 24 line 25), show that the graphical display is a web page on the Internet).

Therefore, it would have been obvious to a person of ordinary skill in the art, at the time the invention was made, to incorporate the teachings of Weinberg into the system of Beatty so that **the visual display is a webpage on the Internet**. The modification would have been obvious because one of ordinary skill in the art would have wanted a convenient, universal way to display the graphical information to the user (Beatty, col. 2 lines 27-30).

As per claim 45, the rejection of claim 1 is incorporated and further, Beatty discloses that **the step of visualizing during the conceptual phase of the system is performed by the visualizer operating in a direct simulation mode before the underlying architecture has been implemented in the development and deployment phases** (col. 3:6-16, "the system of the present invention may include an interface between a processor of the general purpose computer and a real (non-emulated) DSP. In this alternative embodiment, the general purpose computer causes the DSP software to execute within the real DSP (i.e. direct interaction simulation mode), in turn causing the real DSP to change states over time. The general purpose computer monitors the states allowing the architectural display circuitry to update at least one field to reflect changes in the states", and col. 3:21-23, "Additionally, the present invention allows one or more of the user-selectable architectures correspond to DSPs that do not even exist (i.e. during the conceptual phase)").

As per claim 46, the rejection of claim 1 is incorporated and further, Beatty discloses that **the step of visualizing during the development phases of the system is performed by the visualizer operating in a prototype simulation mode** (col. 3:21-23, "Additionally, the present invention allows one or more of the user-selectable architectures correspond to DSPs that do not even exist (i.e. prototype simulation mode)").

As per claim 47, the rejection of claim 1 is incorporated and further, Beatty discloses that **the step of visualizing during the deployment phase of the system is performed by the visualizer operating in an architecture driven monitor mode**(col. 3:60-65, "In an alternative embodiment of the present invention, the architectural display circuitry (i.e. visualizer) allows the user to specify a level of detail regarding the graphical device layout to be displayed in the window (during the architecture monitor mode). The level of detail may be had by zooming in or out or may be had by displaying more or less DSP architecture detail, depending upon the user's wishes").

Response to Arguments

6. Applicants arguments have been considered but they are not persuasive.

In the remarks, p. 2:23-3:19*, the applicant has argued substantially that:*

1) The cited art does not disclose performing a graphical operation on the webpage for dynamic visualization of the graphical elements indicative of the underlying architecture of the software system, specifically that:

a) Beatty does not disclose updating the graphical representation; Beatty only discloses updating a table corresponding to the graphical representation, at 3:11-13;

b) The Beatty visualization is static, in that the graphical representation of the DSP does not change and only a field in a table is changed due to the operation of the system, at p. 3:11-16;

Examiner's response:

1) The examiner disagrees with applicant's characterization of the applied art. The cited art does disclose performing a graphical operation on the webpage for dynamic visualization of the graphical elements indicative of the underlying architecture of the software system, specifically:

a) Beatty discloses updating the graphical representation of the underlying architecture of a software system at col. 9:16-19, "As represented in another higher resolution illustration of the architectural window 410, all of the corresponding states of the registers in the DSP pipeline (i.e. graphical representation of the underlying architecture of the software system), have been updated to represent the latest instructions, (emphasis added)". Beatty further discloses updating the graphical representation in the transition from instruction 28 to 29 as shown in figures 8 and 9. In the architectural display window (top left) of figure 8, the arithmetic logic units (the upside-down trapezoid shaped objects, marked X) have output registers with undefined/empty values. After transitioning to step 29, in figure 9, the output registers of both arithmetic logic units have been updated to show that they are full and have the specific values of pah = 0x1bb0, pal = 0xd9c4, pbh = 0x1bb1 and pbl = 0x8229. Therefore, not only are these values updated in a table, they are also updated in the graphical representation of the software system.

b) The Beatty visualization does change in response to the operation of the system, as described in the response to part a; therefore, the Beatty visualization is dynamic.

Conclusion

7. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Andre R. Fowlkes whose telephone number is (571) 272-3697. The examiner can normally be reached on Monday - Friday, 8:00am-4:30pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Q. Dam can be reached on (571)272-3695. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

ARF

TUAN DAM
SUPERVISORY PATENT EXAMINER